Script-independent Parsing of Text Document Images

Lakshman PRASAD

Abstract-- **We present a general framework for parsing document images into words without restrictions on documents' script or mode of rendering (i.e., handwritten or printed). Our approach is based on the geometric decomposition and metrical characterization of the unprinted/unwritten space (page background) of a document by treating it as a complex shape with multiple "holes" corresponding to connected character sets. This recasts the structure of the unprinted page space in terms of neighborhood relationships and distances among connected character sets. Proximate connected character sets are then grouped into words based on a graph pruning approach. Such a technique is expected to have applications in automatic web browsing and word frequency-based classification of document images on the Internet.**

Index Terms-- **Constrained Delaunay triangulation, document, linguistic, segmentation, shape, string, syntactic**

I. Introduction

The rapidly increasing usage of the Internet as a source of information has resulted in the explosive growth of textual documentation available on the web. While a large portion of such documentation is available in machine-readable (ASCII) form, a significant portion of archived textual information is found in the form of scanned or otherwise imaged documents. While some of this may be eventually converted to machine-readable form, it is infeasible to eliminate this form of information representation altogether in the near future. This is especially true in the case of handwritten documents and documents across languages and scripts. The efficient retrieval of information from document images on the Internet calls for web-based tools that automate query-driven searches for document images. A key aspect of this capability is word spotting and word frequency counting in document images. However, for this to be achieved, it must be possible to automatically isolate individual words from the document. This task is relatively easy when confronted with printed linear text in English, Russian, Italian, etc. However, it is considerably more difficult to do so when presented with handwritten text in these languages or languages with complex script structures such as Arabic, Chinese, and several Indian languages.

Although there have been numerous approaches to the problem of document segmentation [4] [5] [6] [7], they either restrict the variability tolerated in the document (i.e., they expect more or less uniform spacing, regularity of line structure, etc.,) or are applicable to a particular script or format. There have been few attempts at addressing the word segmentation problem across scripts and modes of rendering.

In this paper we present a general technique for parsing document images into paragraphs, sentences, and words without restrictions on documents' script or mode of rendering (i.e., handwritten or printed). Our approach is based on the geometric decomposition and metrical characterization of the unprinted/unwritten space (page background) of a document by treating it as a complex shape with multiple "holes" corresponding to connected character sets. This recasts the structure of the unprinted page space in terms of neighborhood relationships and distances among connected character sets. We will then group proximate connected character sets into words based on a graph pruning approach.

Our research to date on shape analysis has yielded efficient tools to perform document parsing in the above manner: We have developed a novel geometric transform, namely the *Chordal Axis Transform* (CAT) [1] [2], using the Constrained Delaunay Triangulation (CDT) [8] of polygonal shapes. This creates a paradigm for structural segmentation and analysis of complex two-dimensional shapes into morphologically meaningful components. The CAT enables the efficient parsing of shapes into semantically significant feature components. In particular, the CAT segments a shape into two kinds of feature primitives, namely *limbs* and *torsos*. Indeed, every planar shape may be decomposed exclusively in terms of these two kinds of basic feature primitives. The CAT of a shape, moreover, provides information about the metrical attributes and interconnectivity of these primitives. This, then, readily yields a weighted planar graph representation of the shape [2].

We have formulated a linguistic scheme [3] for encoding shapes in terms of the feature primitives that constitute them. The feature primitives obtained by the CAT are represented symbolically, and form the alphabet of the linguistic representation. Symbolic strings represent the sequence of features occurring along the contours of a shape. Further, each character in a string is associated with an attribute vector,

Manuscript received December 7, 2001. This work was supported in part by the U.S. Department of Energy under LDRD-ER Grant No. 2000021.

Lakshman Prasad is with the Los Alamos National Laboratory, Los Alamos, NM 87545, USA (telephone: 505-663-5503, e-mail: prasad@lanl.gov).

which encodes the corresponding feature's metrical attributes, such as length, width, area, etc. Thus, symbolic strings represent the embedding and structure of a shape, while the attributes capture the metrical aspects of features.
In sections II and III we will describe briefly the techniques and tools employed by us to address the problem of text document parsing into words. In section IV we will describe our document parsing algorithm.

II. GEOMETRIC SHAPE FEATURE EXTRACTION

A. Feature Labeling

For a polygonal shape **P**, let CDT(**P**) denote the set of all triangles in its Constrained Delaunay Triangulation [8]. The triangles of a polygon's CDT can be classified into three types, namely those with two external (i.e., polygonal boundary) edges, those with one external edge, and those with no external edges. Each kind of triangle carries morphological information about the local structure of the polygon. Accordingly, they are given different names. A triangle with two external edges marks the termination of a "limb" or a protrusion of the polygon and is called a *termination triangle* or a ***T-triangle***. A triangle with one external edge constitutes the "sleeve" of a "limb" or a "torso", signifying the prolongation of the polygon, and is called a *sleeve triangle* or ***S-triangle***. Finally, a triangle that has no external edges determines a junction or a branching of the polygon, and will accordingly be called a *junction triangle* or a ***J-triangle***. A *limb* λ is a chain complex of pairwise adjacent triangles, of the form ***TS . . . SJ*** or ***JS . . . ST*** (Fig.1), and a *torso* τ is a chain complex of pairwise adjacent triangles, of the form ***JS . . . SJ*** (Fig.2).

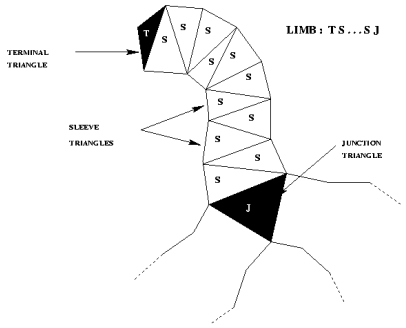


Fig.1 A limb chain complex

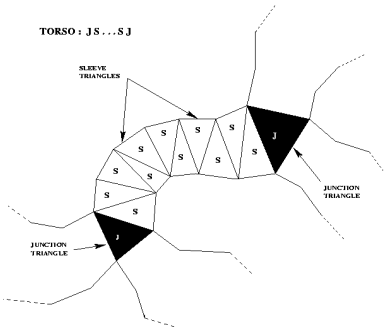


Fig.2 A torso chain complex

The number of sleeve triangles in a limb or a torso is allowed to be zero; thus, the duos ***JT*** or ***TJ*** also define limbs and, likewise, the duo ***JJ*** also defines a torso. Torsos can be further distinguished into two categories: If all the internal edges of the sequence of ***S***-triangles between the two ***J***-triangles of a torso connect pairs of points that belong to the same connected contour component, then the torso is termed a *stem*. Otherwise (i.e., even if one internal edge of the sequence of ***S***-triangles connects a pair of points that belong to different connected contour components,) it is termed a *handle*. If there are no ***S***-triangles between the two ***J***-triangles of a torso then the above conditions apply for the common edge of the two ***J***-triangles. Both sides of a stem can be accessed by traversing along a connected contour component of a shape, while only one side of a handle is accessible to any connected contour component. It is easy to see that handles occur only in shapes that have at least one hole.
The limbs, stems, and handles of a shape form its generic feature primitives. Each feature primitive is assigned a vector ν of attributes, which may have the length, width, variance, area, etc., of the feature primitive as its components[3]. These components serve to capture the "vital statistics" of the feature primitive.

B. Shape Skeletonization and Pruning

A skeleton of the polygonal shape that has the same connectivity as the shape and serves as the local axis of symmetry of the shape can be constructed by making only local constructions within each triangle in the shape's CDT as follows (Fig. 5):

1) In each ***S***-triangle join the midpoints of the edges that are internal to the shape (i.e., not lying on the shape boundary).
2) In each ***J***-triangle:
(a) join the midpoints of all the sides of the ***J***-triangle to its circumcenter (the intersection of the perpendicular bisectors of the sides of the triangle) if the triangle is acute (i.e., if the circumcenter lies inside the triangle); or
(b) join the midpoint of the longest side of the triangle to the midpoints of the other two sides if the triangle is not acute (i.e., if the circumcenter lies outside the triangle).

The skeleton induces a planar graph representation of the shape structure. Indeed, consider the graph whose vertices are nodes of degree 3 (junctions) or degree 1(terminations) in the skeleton, and whose edges are the polygonal arcs connecting nodes of degrees 3 and 1. Thus an edge between two degree 3 nodes represents a torso (i.e., a stem or a handle), while an edge between a degree 3 and a degree 1 node represents a limb. By weighting each edge of the graph with the metrical information of the corresponding feature, we obtain an attributed graph representation of the shape's structure.
The shape and its skeleton (and therefore its shape graph) can be further pruned to excise undesirable morphological features. In the CDT of a polygonal shape, each side of a ***J***-triangle that connects boundary points of the same boundary component subtends a chain of polygonal vertices that does not include the vertex of the ***J***-triangle opposite to this side (Fig.3). The ratio of morphological significance $\rho = d / |AB|$, of the distance *d* between the farthest point *p*

of the chain from the side *AB*, of the junction triangle *ABC*, is a quantitative indication of the importance of the portion *AopqrsBA* in describing the overall shape (Fig.3).

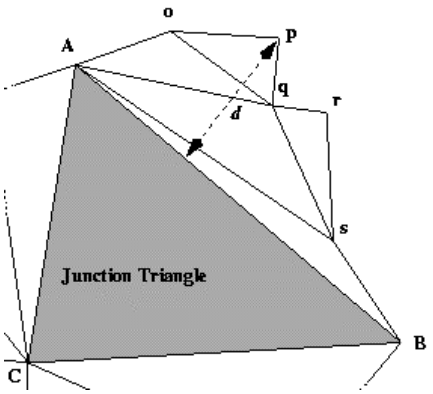


Fig.3. Pruning a morphologically insignificant feature. The contour points A and B must belong to the same contour component.

Whenever a part of a shape (subtended by an edge of a ***J***-triangle) is morphologically insignificant, i.e., whenever ρ is less than some threshold, the part is excised from the shape. The edge subtending the excised part becomes part of the new polygonal boundary (i.e., *A* and *B* become neighboring boundary vertices of the modified polygon), while the ***J***-triangle to which the edge belongs becomes an ***S***-triangle. This results in a simplified shape that still represents the salient features of the original shape. Accordingly, the new shape's skeleton does not reflect the morphologically insignificant branches associated with the excised part of the shape (Fig. 4).

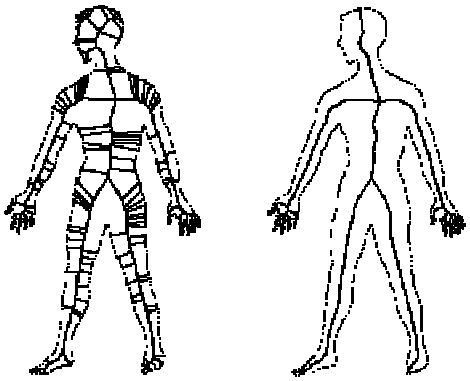


Fig.4. Polygonal human shape with noisy boundary shown with skeletons before and after (ρ=0.4) pruning.

III. SYNTACTIC SHAPE CHARACTERIZATION

We are now in a position to encode the exterior of a polygonal shape via a syntactic string of feature primitives:

Tracing (counterclockwise, say,) the outer contour of a polygonal shape, which has been decomposed into its feature primitives via its CDT, we will encounter, in sequence, the feature primitives (i.e., limbs, stems, and handles) of the shape that are adjacent to its outer contour (Fig. 6). If we encounter a limb, we will record this by appending the symbol " ***l*** " to the string (which is initially empty), if we encounter a handle, we will record this by appending the symbol " ***h*** " to the string. Finally, if we encounter a stem, we will record this by appending the symbol " ***(*** ", if this is the first time this stem has been encountered, or by the symbol " ***)*** ", if this is the second time the stem has been encountered (Fig. 6). The resulting string is a sentence in a language that characterizes the exteriors of shapes in terms of features occurring along their outer contours. The symbols ***l***, ***h***, ***(***, and ***)*** are the terminal characters of the language. Each symbol in the string is associated with the attribute vector ν = ***(***λ, ω, α***)*** of the corresponding feature, where λ is the length, ω is the average width, and α is the area of the corresponding feature primitive (Fig. 7). These attributes can be computed from the CAT, as can other attributes such as directionality, mean curvature etc., as the application demands. The attributes may further be normalized to achieve scale-free representations if necessary. Thus, we have an attributed syntactic representation of polygonal shape exteriors.

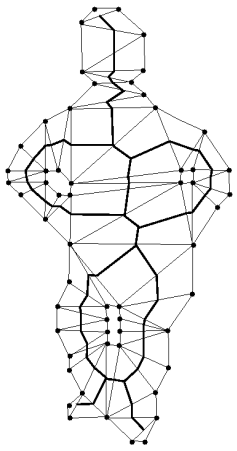


Fig.5 Feature primitives highlighted by the skeleton

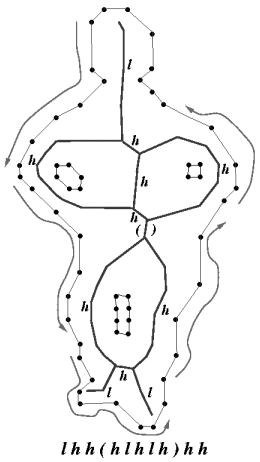
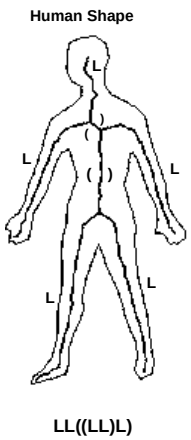


Fig.6 Syntactic feature labeling and string representation of shape exterior



Feature	α	λ	ω
L	202.312	20.524	9.857
L	286.204	41.111	6.962
(	49.714	4.590	21.664
(	165.397	20.480	16.152
L	366.757	50.986	7.193
L	333.293	45.503	7.325
)	165.397	20.480	16.152
L	274.325	39.921	6.872
)	49.714	4.590	21.664

α = Area of the feature primitive
λ = Length of the feature primitive
ω = Mean width of the feature primitive

Fig.7 Syntactic representation of a human shape and the corresponding feature attributes

This representation is however not unique. Indeed, consider a string $\Sigma = C_1(C_2)C_3$. An equivalent representation of this string is $\Sigma = C_2(C_3C_1)$. This is because the outer contour of a shape is a closed curve, and as such, there is no intrinsically distinguished point on it from which we can start scanning for feature primitives. Thus, depending on where we start tracing on the outer contour, we get different representations of the same string. One way to normalize string representations would be to lexicographically order all possible rotations, and pick the first string. Indeed, giving a definite order to the alphabet of terminals induces the lexicographic ordering. Elsewhere [3] we prefer to use a normalization that has value from the point of shape recognition. For the purposes of this paper, however, the exact method of normalization of the string is irrelevant.

In many applications, the syntactic characterization of the exterior of a shape, without regard to holes, may be sufficient. A complete description of a shape with holes is also possible with the method of syntactic encoding of shape exteriors that we have introduced. In essence, this complete representation is obtained by encoding each inner hole contour via the feature primitives occurring along it. The shape may then be syntactically represented by a set of strings, each for one hole, and one string for the outer contour. The outer contour will be given the number 0 and the holes are numbered randomly with numbers 1 to *n*, where *n* is the number of holes. In shape recognition applications, an unambiguous representation of shapes with holes is necessary. Elsewhere [3], we have developed a technique for the canonical ordering of the strings of a shape, so that two identical shapes have their strings listed in the same order. We will however not describe the technique here as it is not necessary to have a canonical representation for our present purposes, although our illustrations reflect this canonical ordering. Thus, we have a complete attributed syntactic representation of a shape with *n* holes in terms of *n*+1 strings.

IV. Segmenting a Document into Words

For the purposes of this paper, we will assume that a digitized document image has been segmented into a binary image, with the unprinted portions of the document in white and the text in black. Indeed, segmentation of documents is far easier than that of general imagery, and there are several methods that give satisfactory results. We will treat the white portions (shown in green in our illustrations) of the segmented image as the interior of a complex shape with the black (text) portions not intersecting with the document boundary as holes in the shape (Figs. 8b & 9b). If the shape has multiple connected components (say, due to a line running across the entire document), we will treat each connected component separately, since each such component corresponds to a portion of the page. Finally, we will ignore or discard all connected shape components that do not have any holes in them, as these correspond to portions of the document with no text in them. This step eliminates from consideration all "islands" of page created by loops or enclosures in text (as found in "A", "b", "g", etc.) For each retained connected shape component, all (i.e., outer and inner) polygonal contours are extracted (red lines in Fig. 8c). At this stage, very short contours (4 to 6 pixels) corresponding to document noise are suppressed, and the remaining contours are subjected to Gaussian smoothing. A CDT of the interior of each component is performed, and the various shape features (i.e., limbs, stems, and handles) are identified and metrically labeled with the average width attribute (the area and length attributes are not relevant for our analysis here) via the CAT. The CAT of the shape is then pruned to eliminate all limbs and stems, so as to retain only handles. This is done because we are interested only in the separation between connected character components (i.e., holes) and not in the involutions of these holes. This pruning is equivalent to setting the ratio of morphological significance $\rho = \infty$. The results of these operations are illustrated in Fig. 8c, where the contours are shown in red, the triangle edges in light blue, the sleeve triangles in yellow, the junction triangles in white, the pruned areas in green, and the pruned skeletons consisting only of handles in dark blue.

The inner hole contour strings are numbered, shown in yellow in Fig. 8e. As a result, each handle obtains a pair of numbers, each corresponding to a contour bounding the handle. This handle-contour association is important for building the lookup table for character adjacency graph construction and pruning.

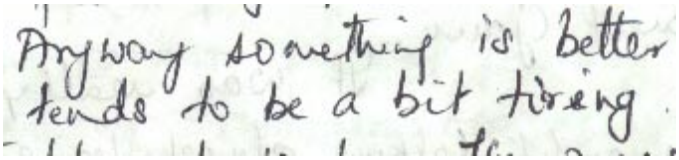


Fig. 8a Sample portion of a handwritten English text document image

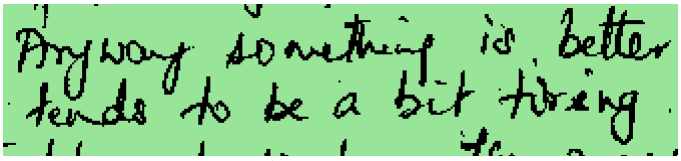


Fig. 8b Binary segmentation of document into character groups and page

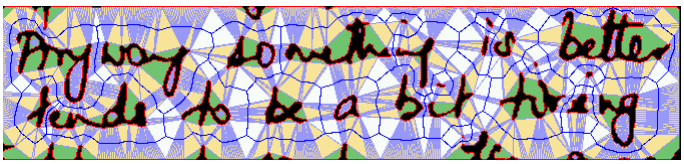


Fig. 8c Constrained Delaunay triangulation (light blue) with pruned skeleton (dark blue) consisting of only handles. Pruned areas are shown in green, sleeve triangles in yellow, junction triangles in white, and smoothed contours in red.

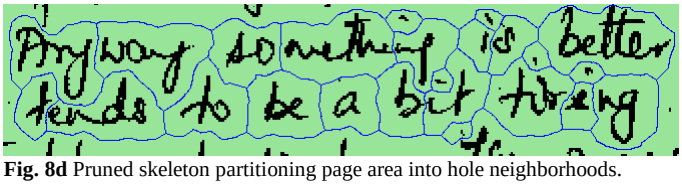


Fig. 8d Pruned skeleton partitioning page area into hole neighborhoods.

Fig. 8e Numbering of hole contours. The associated hole neighborhoods also inherit the numbering of the hole contours they surround.

1		8		14		19	
21	20.805	7	22.185	13	18.713	18	13.776
5	13.819	20	13.507	24	8.505	20	21.963
4	16.643	18	8.294	11	17.088	20	
2	6.91	9	4.661	23	12.437	19	21.963
2		9		15	25.678	18	3.746
1	6.91	8	4.661	15		8	13.507
4	13.057	18	7.929	14	25.678	7	19.229
3	15.217	22	10.662	23	14.343	6	22.145
3		18	19.753	11	15.647	21	6.752
2	15.217	17	15.882	9	23.253	21	
4	3.817	15	23.253	17	4.479	20	6.752
4		11	14.06	16	10.805	6	15.326
3	3.817	10	11.434	16		5	16.496
2	13.057	10		15	10.805	1	20.805
1	16.643	9	11.434	17	5.465	22	
5	17.986	11		17		9	10.662
5		9	14.06	16	5.465	18	6.46
4	17.986	15	15.647	15	4.479	23	
1	13.819	23	15.823	9	15.882	11	15.823
21	16.496	14	17.088	18	24.515	15	14.343
6	18.881	24	14.484	18		14	12.437
6		12	5.798	17	24.515	24	
5	18.881	12		9	19.753	11	14.484
21	15.326	11	5.798	22	6.46	14	8.505
20	22.145	24	8.452	9	7.929	13	10.336
7	16.936	13	9.664	8	8.294	12	8.452
7		13		20	3.746		
6	16.936	12	9.664	19	13.776		
20	19.229	24	10.336				
8	22.185	14	18.713				

Fig. 8f Lookup table showing contour number (red) with contour numbers of adjacent hole neighborhoods (blue) and the corresponding average widths of handles separating two contours.

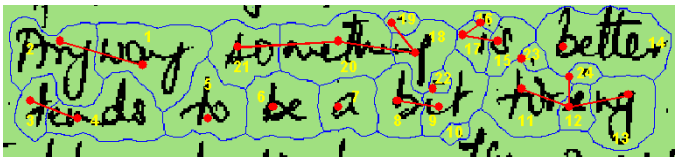


Fig. 8g Pruned character adjacency graph showing subgraphs representing word groupings based on average handle widths in the lookup table

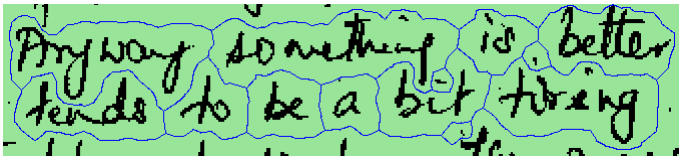


Fig. 8h Merged hole neighborhoods reflecting words segmented based on the pruned character adjacency graph

We will associate the connected page region surrounding each text hole, and enclosed by a fence of handles, with the hole for convenience. These will be termed *hole neighborhoods*. Thus, hole neighborhoods partition the page space into disjoint neighborhoods of connected character groups (Fig. 8d). We will generate a connected planar graph termed the *character adjacency graph* (CAG), which is dual to the pruned skeleton graph, to represent the neighborhood relationships between connected character components. Each hole neighborhood is represented by a vertex of the CAG, and two vertices are joined by an edge if the corresponding hole neighborhoods share a handle. Each edge of CAG is weighted by the average width attribute of the corresponding handle. The numbering of the contour strings produces a lookup table (Fig. 8f) consisting of a list of contour numbers (numbers in bold red) and their adjacent contours' numbers (numbers in bold blue) with corresponding average widths (in number of pixels, given by numbers in black in the table) of the handles separating them. Only hole contours and their mutual adjacencies are shown in the lookup table as we are not interested in the adjacency relationships between the document's outer contour (numbered 0) and the hole contours.

We use a simple rule to prune the CAG generated above, in consultation with the lookup table and handle directionalities:

An edge is deleted in the CAG if and only if

i) it does not represent the narrowest separating handle for at least one of the two hole neighborhoods corresponding to the vertices upon which the edge is incident, or

ii) the width of the handle it represents is greater than the average of the minimum average handle widths of all contours.

For example, the minimum average handle width of contour 16 is 5.465 pixels (separating it from contour 17), and that of contour 17 is 4.479 pixels (separating it from contour 15) as displayed in the lookup table in Fig. 8f. In the case of the written English script (Fig. 8a), the average minimum handle width of all contours is 7.872 pixels.

The pruning operation results in several connected subgraphs of the CAG with mutually exclusive vertex sets (Fig. 8g). These graphs yield the desired grouping of character sets to form words (Fig. 8h), according to the pruning rules employed. In the example of the English script, the number of distinct connected character groups is reduced from twenty four (Fig. 8e) to twelve word groups (Fig. 8h), of which two are punctuations or modifier strokes, and may be discarded on account of the negligible areas of their holes.

V. Discussion

The graph-pruning rule we have employed here is a heuristic based on common knowledge about grouping English handwritten characters into words. This rule is by no means complete or sufficient to parse most generic handwritten English documents. The formulation of a proper set of pruning criteria is a subject of future research and experimentation. Neural network-based approaches may be appropriate to precipitate such rules by training on large sample classes and using handle attributes inputs. It may also be necessary to also compute attributes other than of handles, such as rough orientation (i.e., whether horizontal, vertical, or diagonal), position on page, etc, for more sophisticated character grouping and assignment operations. Indeed, directionality of handles is important for segmenting lines of text, as words along a line tend to be separated by vertical torsos. Similarly, horizontal torsos separate lines and paragraphs, albeit of different width classes.

It is also possible that the grouping rules differ from script to script. For example, our experiment with Arabic printed text

(Fig. 9a) shows that pruning the CAG based on average widths of handles (Fig. 9e) leads to erroneous word formations, as evidenced by the grouping of character holes twelve and twenty-one in Fig. 9f. However, when sub-rule i) is applied without using sub-rule ii) employing minimum handle widths (not shown in table) instead of average handle widths, we obtain a more correct grouping into words (Fig. 9g). If, on the other hand, we had used minimum handle widths instead of average handle widths in the case of the handwritten English script example, we would have introduced groupings between character sets in the top and bottom rows due to the descending strokes and ascending strokes interacting with lower and upper rows, respectively.

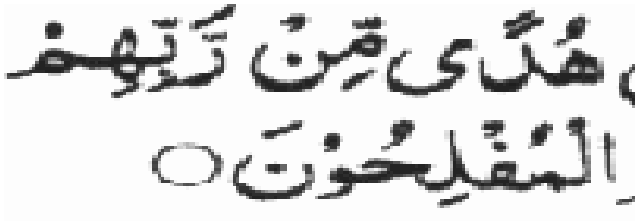


Fig. 9a Sample portion of a printed Arabic text document image

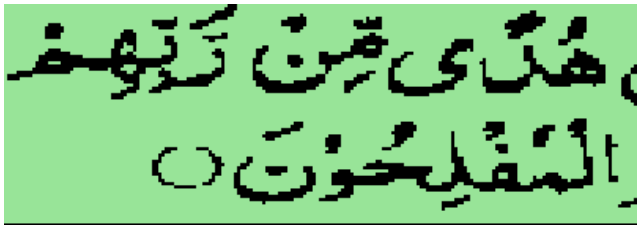


Fig. 9b Binary segmentation of document into character groups and page

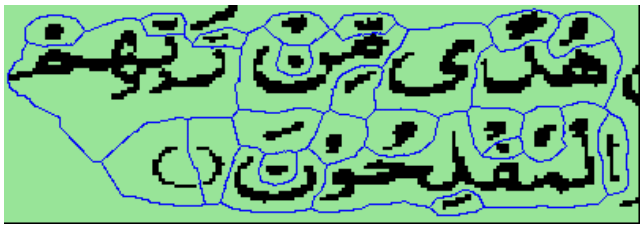


Fig. 9c Pruned skeleton shown partitioning page area into hole neighborhoods

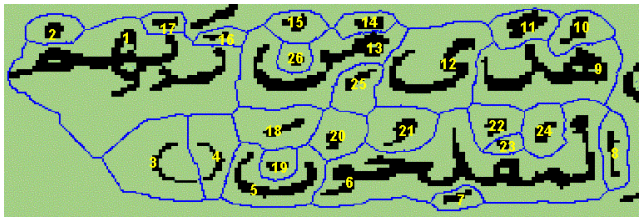


Fig. 9d Numbering of hole contours.

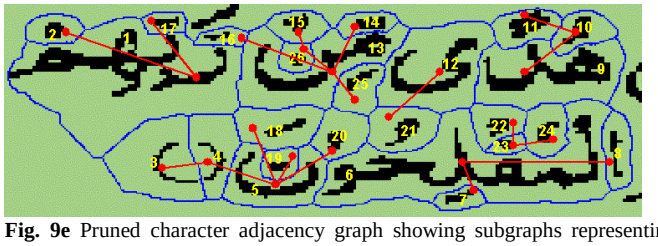


Fig. 9e Pruned character adjacency graph showing subgraphs representing word groupings based on average handle widths in the lookup table.

Fig. 9f Merged hole neighborhoods reflecting words segmented based on the pruned character adjacency graph in Fig. 9e.

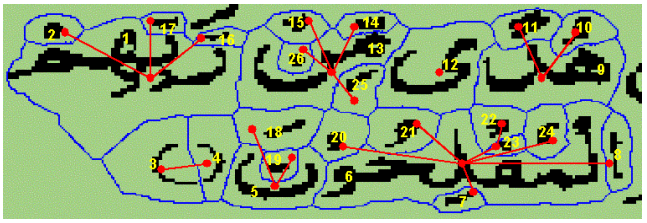


Fig. 9g Pruned character adjacency graph showing subgraphs representing word groupings based on minimum handle widths (not shown in lookup table).

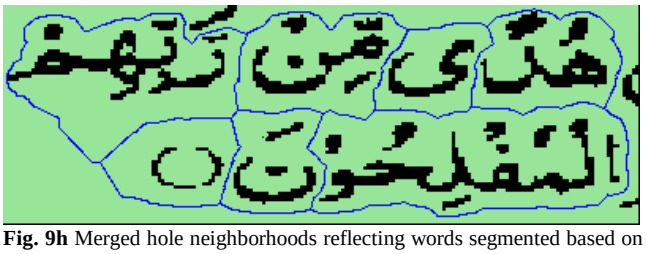


Fig. 9h Merged hole neighborhoods reflecting words segmented based on the pruned character adjacency graph in Fig. 9g.

1		7		14		22	
17	8.161	6	3.927	13	4.866	6	10.464
16	12.085	8		15		12	15.802
13	10.203	6	5.849	13	10.128	9	9.621
18	23.664	9		26	7.722	24	9.848
4	19.522	6	11.267	13	9.204	23	3.187
3	21.823	24	14.042	16		23	
2	5.108	22	9.621	13	11.025	6	8.298
2		12	7.014	1	12.085	22	3.187
1	5.108	11	10.93	17		24	8.837
3		10	6.709	1	8.161	24	
1	21.823	10		18		6	10.691
4	11.206	9	6.709	1	23.664	23	8.837
4		11	8.213	13	11.147	22	9.848
3	11.206	11		20	9.218	9	14.042
1	19.522	10	8.213	5	12.012	25	
18	16.718	9	10.93	19	9.039	12	7.052
5	6.02	12	12.049	5	8.009	21	18.319
5		12		4	16.718	20	14.322
4	6.02	11	12.049	19		13	6.163
18	8.009	9	7.014	5	6.692	26	
19	6.692	22	15.802	18	9.039	13	6.701
18	12.012	6	13.889	20		15	7.722
20	7.728	21	9.295	5	7.728		
6	6.04	25	7.052	18	9.218		
6		13	5.492	13	15.703		
5	6.04	13		25	14.322		
20	9.405	12	5.492	21	15.796		
21	10.567	25	6.163	6	9.405		
12	13.889	20	15.703	21			
22	10.464	18	11.147	6	10.567		
23	8.298	1	10.203	20	15.796		
24	10.691	16	11.025	25	18.319		
9	11.267	15	9.204	12	9.295		
8	5.849	26	6.701				
7	3.927	15	10.128				
		14	4.866				

Fig. 9i Lookup table showing contour number (red) with contour numbers of adjacent hole neighborhoods (blue) and the corresponding average widths of handles separating two contours.

The examples provided by us here serve to illustrate how the general geometric framework proposed by us may be used to efficiently segment and extract words from text document images across different languages and in both printed and hand-written documents. The derivation of other useful CAG edge attributes and pruning rules is outside the scope of this paper.

The overall average time complexity of our algorithm is of the order $n\log n$, where n is the number of contour points of the segmented unwritten page shape. This is due to the fact that the average time complexity of the CDT is of the order $n\log n$.

VI. Conclusion

We have presented here a broad but versatile geometric framework for text document image parsing into words. Successful implementation of this technique with adaptations to specific scripts could significantly improve over existing techniques for document parsing, and enhance the robustness of word spotting and word frequency algorithms. Our continuing research in this area will focus on computing more sophisticated handle descriptors, and formulating better heuristic graph-pruning rules for character grouping.

Acknowledgment

This work is supported by U.S. DOE LDRD-ER grant #2000021.

References

[1] L. Prasad, "Morphological analysis of shapes," *CNLS Newsletter*, No. 139, July 1997, LALP-97-010-139, Center for Nonlinear Studies, Los Alamos National Laboratory.

[2] L. Prasad, R. L. Rao "A geometric transform for shape feature extraction," *Proc. of the 45th SPIE Annual Meeting*, **4117** Vision Geometry IX, San Diego, CA, 2000.

[3] L. Prasad, A. N. Skourikhine, B. R. Schlei, "Feature-based syntactic and metric shape recognition," *Proc. of the 45th SPIE Annual Meeting*, **4117** Vision Geometry IX, San Diego, CA, 2000.pp 234-242.

[4] R. Manmatha and W. B. Croft, Wordspotting: Indexing Handwritten Manuscripts, in *Intelligent Multimedia Information Retrieval*, ed. Mark Maybury, AAAI/MIT Press, 1997.

[5] I. S. I. Abuhaiba, M. J. J. Holt, and S. Datta "Recognition of off-line cursive handwriting," *Computer Vision and Image Understanding*, Vol 71, No. 1, July, pp. 19-38, 1998

[6] F. Wahl, K. Wong, and R. Casey "Block segmentation and text extraction in mixed text/image documents," *Computer Vision, Graphics, and Image Processing* 20: pp 375-390. 1982

[7] D. Wang, S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision, Graphics, and Image Processing* 47: pp 327-352, 1989

[8] J. R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," *Association for Computing Machinery, First Workshop on Applied Computational Geometry* Philadelphia, PA, pp 124-133, , May 1996.